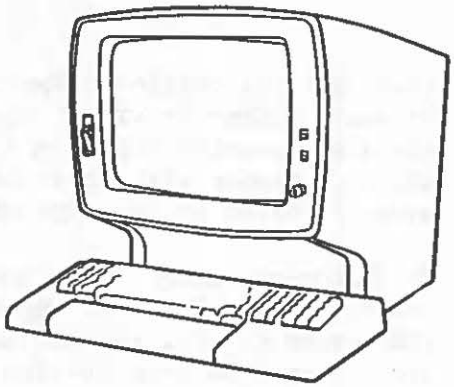
1982-08-27

Computer Center News Letter / August 27, 1982

Monterey, California, Naval Postgraduate School

Computer Center News Letter



NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

August 27, 1982
Volume 14, No. 6

TABLE OF CONTENTS *PAGE*

* * *

MVS TO CMS COMMUNICATION

MVS ("batch") users no longer need to stand by the system printer waiting for the output of batch jobs. Your results can be reviewed at your terminal and hard-copy avoided. The newly installed RSCS/NET (Remote Spooling Communications Subsystem/Networking) facility provides MVS users with the ability to return output from MVS jobs to their CMS virtual readers, A-disks or temporary disks. As a file on a CMS disk, the output can then be browsed, printed, edited, or erased. To set up this return of output requires an additional Job Control Language (JCL) card. After the JOB card, add the following line:

```
//*MAIN ORG=NPGVM1.nnnnP
```

where "nnnn" is your user member.

When the job completes execution on MVS, the output will be sent back to the virtual reader of VM/CMS user "nnnnP". To place this file on a CMS disk, use the CMS function RDR. It is also possible to examine a file waiting in the virtual reader without transferring it to the CMS disk and consuming valuable space. To do so, use the CMS function RLOOK.

In addition, after a job has been submitted to MVS, the user can issue the CMS command INQ which will report the job's progress towards completion. A second CMS command, CAN, is available for cancelling an unnecessary job. Both of these commands have on-line documentation obtainable by typing INQ ? or CAN ?

Further information on RSCS/NET is available in the Computer Center Technical Note MVS-06, Use of RSCS/NET for MVS and CMS Communications. This document shows how to send part of the output to the terminal and part to the system printer. Copies should be available in the Consulting Room, In-146, by the time this Newsletter is published.

* * *

FILEMASTER CAN HELP YOU

Prof. Arthur Schoenstadt (Department of Mathematics) has written a package of VM/CMS EXEC2 routines named FILEMASTER and made it available to other users. It may take the pain out of using many of the common MVS (batch) file handling utilities. One basic concept behind FILEMASTER is the use of selectable menus, together with limited English language input, e.g. data set names. FILEMASTER automatically generates MVS job control language to accomplish many functions such as allocating space on disks or MSS, moving CMS files to MVS, copying libraries or members between devices, compressing data sets, listing Volume Tables of Contents, scratching data sets or members, renaming data sets, etc.

Complete information on use of FILEMASTER can be obtained by issuing

FILEMASTER ?

(You will also be given the opportunity of obtaining a printed copy of the documentation.)

* * *

DISSPLA AND TELL-A-GRAPH DOCUMENTATION

Reference and circulating copies of the complete DISSPLA and TELL-A-GRAPH manuals are now available in the Center's Administrative Office, In-131. Copies of the DISSPLA First Facts booklet and the Pocket Guide can also be obtained for reference or circulation. The NPS bookstore has ordered copies of the DISSPLA Pocket Guides. They will be for sale at a very nominal price.

FILES USED BY DISSPLA

The DISSPLA graphics system uses certain files in its execution. The corresponding file unit numbers are reserved by DISSPLA and therefore should not be used in any FORTRAN program which calls the DISSPLA package. The reserved files differ somewhat depending on the operating system used.

On VM/CMS, the timesharing system, the DISSPLA exec automatically assigns FILEDEFS for units 11 through 18. (Some of these may not be used as they are allocated dynamically when needed, but it is wiser for the user to choose other files for program I/O.) The DISSPLA exec also assigns units 05 and 06 for terminal I/O. Often the user wishes to change these files to refer to a disk file for input (05) or the printer for output (06). This is easily done by invoking the self-prompting version of the DISSPLA exec, i.e., by simply typing

```
DISSPLA
```

at the terminal. (Type "DISSPLA ?" for more information if you are not familiar with this exec.)

On MVS, the batch operating system, both DISSPLA and the Versatec plotter reserve files 05 through 08, and files 11 through 18. Of course, files 05 (card input), 06 (printer output), 07 (punched card output), and 08 (alternate plotter output) may be overridden, when desired, by following the standard rules for overriding DD statements in a cataloged procedure as defined in section 5.3.2 of the Center's Technical Note MVS-01.

* * *

1982 BIMED PACKAGE AVAILABLE

The 1982 version of the BMDP Statistical Software package will be available at NPS on 1 September 1982. The two cataloged procedures affected are BIMED and BIMEDT. The 1981 version which will remain on the system for 90 days can be invoked via cataloged procedures BIMED81 and BIMEDT81. If no problems are found with the 1982 version, the older version will be erased around 1 November.

The BMD organization reported the following changes in the 1982 version:

1. BMDP1T, Spectral Analysis, is being presented for the first time.

2. A completely new algorithm for Block Clustering is implemented in the experimental program, BMDQ3M.

3. The prototype program, BMDPYD, is available for easily embedding new analyses into a BMDP program.

4. BMDP1F, BMDP2F and BMDP3F print notices that BMDP4F (Two-Way and Multiway Frequency Tables) is available for all contingency table analyses.

The basic documentation remains BMDP Statistical Software 1981 Edition, University of California Press, ISBN 0-520-04408-8. Reference copies are available in the Consultation Office, In-146, and the Computer Center Library, In-162.

* * *

LARGE PRINTER OUTPUT - ADDITIONAL INFORMATION REQUIRED

If you have a program which generates more than 10,000 lines of printer output, you must submit a "job information" card to the operator. These cards may be found in In-140 on the counter near the output distribution boxes.

* * *

HELPFUL HINTS ABOUT THE 3278 KEYBOARD

(This article is adapted from one in a recent issue of the Virginia Polytechnic Institute Newsletter)

1. The PA1 key can be used to enter CP mode. In CP mode only CP commands can be entered. This key can be used, for example, while in FLIST to enter the CP command PURGE. To return to FLIST, B (for begin) would be entered.

2. The PA2 key used in CMS mode clears everything on the screen except what is in the user input area. Therefore, if you have typed in a command (and not yet hit the return key) and realize that you will not have enough room on your screen to display the results of the command, you can hit the PA2 key which will clear the screen except for the command and then hit the RETURN key. This prevents you from having to clear the screen with the CLEAR key and re-type the command.
   The PA2 key can be used in FLIST to enter the CMS subset. While in the CMS subset the user can enter a limited number of the CMS commands. RETURN should be typed to get back into FLIST.

3. The ERASE EOF key is used to delete everything on either a command line or the part of the line in a file to the right of and including the character where the cursor is located. So, for example, if the command:

   XEDIT TEST FILE

   were on the command line with the cursor positioned on the 'F' and the ERASE EOF key pressed the line would look like:

   XEDIT TEST _

   with the cursor remaining in the same position.

4. The ERASE INPUT key is used to delete the entire input area regardless of where the cursor is positioned. If the command:

   XEDIT TEST FILE

   were being entered and the ERASE INPUT key pressed the entire command would be deleted.

   NOTE: This key should be avoided when in XEDIT, since it causes the entire screen to be cleared.

5. The INSERT key (â) is used to insert characters either in a command line or in a line in a file being edited. If the command:

   XEDIT TEST FLE

   were entered on the screen with the cursor positioned under the 'L' and the INSERT key pressed, an 'I' could be typed in and would be inserted before the 'L'. Thus the line would read:

   XEDIT TEST FILE

   The RESET key must be pressed to get out of insert mode.

* * *

VM MANUALS ORDERED FROM IBM MAY BE PREMATURE

If you order IBM manuals for VM at their Webster Street (Monterey) office or by mail, you will now get editions that are valid for VM/SP Release 2. We are still using Release 1 at NPS. It will probably be six to nine months before we switch over. Therefore we advise you to preserve your present VM/CMS manuals carefully. The new manuals are presently useful only in describing future changes.

* * *

DISPLAY MANAGEMENT SYSTEM (DMS) TECHNICAL NOTE

The Display Management System (DMS/CMS) is an IBM product which allows applications programs to utilize full-screen input and output on IBM 3270 series terminals. The Center has recently published a User's Guide to DMS for Assembler Language, Fortran and CMS EXEC programmers. It is numbered TN VM-09 and copies are available in the Consultation Office, In-146.

USING FILEDEFS

(The following article is adapted from one appearing in the MIT computing newsletter.)

Every operating system has some way of defining files--that is, associating each file in your catalog or directory with a reference name or unit number in your program. MVS has DD statements; in CMS, FILEDEF statements perform this function.

Suppose, for example, that you have a Fortran program that reads data entries from a file called INPUT DATA into an array called TABLE, sorts the data, and writes the sorted data to a file called OUTPUT DATA. In your program, you decide to assign the input file to unit 1 and the output file to unit 2. Your program reads in data with the statement:

```
READ (1,100) (TABLE (I), I=1,50)
```

and writes the data to the output file with the statement:

```
WRITE (2,100) (TABLE (I), I=1,50)
```

In both statements, the second argument (100) refers to the statement number of the format statement, which specifies the way the data will be read or written. The first arguments (1 and 2) are unit numbers to be associated with the input and output files respectively. You make the association by issuing FILEDEF statements outside the program, at CMS command level. The simplest FILEDEF statements for our FORTRAN example (assuming the files are stored on the A disk) would be:

```
filedef 1 disk input data

filedef 2 disk output data
```

Options on the FILEDEF statement allow you to specify file characteristics such as record format, record length, and block length. The most important options are:

RECFM a	record format of file; 'a' can be F (fixed length), V (variable length), etc.
BLOCK nnnnn	logical block size of file in bytes.
LRECL nnnnn	logical record length of file in bytes.

It is generally a dangerous practice to rely on default values or conditions of file attributes when using FILEDEFs. Specify these attributes explicitly to avoid error messages like "OPEN (or input or output) ERROR ON FILE..." or "DISK FULL". You'll also avoid output files that are padded or contain only selected input records. For a Fortran program, if you have specified a file attribute incorrectly, you are likely to receive this error message:

```
IHO219I FIOCS - MISSING DD CARD OR DCB ERROR FOR ASCII TAPE...
```

For most files to be processed by Fortran formatted I/O, or XEDIT, fixed-length records with a record length of 80 characters will suffice. Therefore, your FILEDEF can be:

```
filedef 1 disk input data a (recfm f lrecl 80
```

If you are reading an existing file and are not sure of its characteristics, you can find out by issuing the LISTFILE command for that file with the ALLOCATE option. For example, if your file's name is PROG LISTING A, type

```
listfile prog listing a (alloc
```

and the system responds with file information in the following form:

```
FILENAME FILETYPE FM  FORMAT LRECL RECS BLOCKS
PROG     LISTING  A1  V      121   6    1
```

Therefore, your FILEDEF can be:

```
filedef 1 disk prog listing a (recfm v lrecl 121
```

FILEDEFS are cleared as soon as a program is executed or compiled, when a run-time error is encountered, or when HX is used to halt program execution. They are also cleared when you log off or when the system crashes. If you are in doubt as to what FILEDEFs are presently in effect, simply type QUERY FILEDEF. Fortran files will be listed next to their system-defined names, FTnnF001, where nn is the unit number that appears in your program.

If you are working on a program that is to be recompiled and re-executed often, it can be annoying to have to keep issuing FILEDEFs. You can avoid this by using the PERM option of the FILEDEF statement. For this Fortran example you would type the following for your input file FILEDEF:

```
filedef 1 disk input data a (recfm f lrecl 80 perm
```

Files defined using the PERM option stay in effect until you specifically clear those definitions, redefine them, or log off; or when your program terminates with an error, or as a result of an HX command.

FILEDEFs for particular Fortran unit numbers can be changed (to change the input data file, for example) simply by issuing a new FILEDEF statement. You can also clear existing FILEDEFs by using the CLEAR option. Issuing for example,

```
filedef 1 clear
```

clears any existing association of unit 1 with any file. You can clear all existing FILEDEFs by typing:

```
filedef * clear
```

If the PERM option is not used, it is important to issue the FILEDEFs after compiling but before loading your program. The act of compiling destroys existing FILEDEFs. If you load and run your Fortran program without issuing the needed FILEDEF for the input file, your program will terminate with an error message like this one:

```
IHO218I FIOCS - I/O ERROR 120S INPUT ERROR 001 on FT01F001
```

Here FT01F001 is a default filename supplied by the system, which is associated with the unit 01 (the two-digit number following FT) that you referenced in your READ statement. If you neglect to issue a FILEDEF for the output file, your

program will run to completion and the system will create an output file with the default name "FILE FTnnF001 A1", where nn is the unit number appearing in your write statement.

Suppose you issue a FILEDEF for the input file but simply misspell the name of the file. In that case, your Fortran program terminates with the message:

IH0900I FIOCS - END OF DATA SET ON UNIT 1

assuming again that unit 1 appears in your READ statement.

* * *

RECORD BLOCKING AND SYSTEM PAGE SIZE

One of the reasons for blocking records is to make the optimum use of the storage space allowed. In a previous Newsletter, we discussed a good common blocksize to use when data sets are to be moved between 3330 and 3350 disk packs. This article deals with record blocking and our system's memory page size, 4096 bytes.

If a block of instructions, and/or data is read into memory (either by the system or a program READ), and that block is larger than 4K bytes, the system has to 'page' that block. This means that when the first 4K page is full, the system will allocate another 4K page, if the memory is available. If it is not, then the first page is written to external page storage before the remainder of the block can be read in. Some system overhead is involved whenever paging is done.

This means that, in general, users should choose blocksizes that are somewhat less than 4096.

* * *

RECENT ADDITIONS TO THE COMPUTER CENTER LIBRARY

Books

Author	Title
Warlick, Charles H.	1981-82 Directory of Computing Facilities in Higher Education

Proceedings

Author	Title
ACM	1981 ACM '81 Conference Proceedings
ACM	1981 Seventh Data Communications Symposium
ACM	1982 Proceedings of the Sigplan '82 Symposium on Compiler Construction

The Newsletter appears semi-quarterly and is written by members of the staff, W. R. Church Computer Center (code 0141), Naval Postgraduate School, Monterey, CA 93940. Requests for further information or suggestions for articles for the Newsletter may be addressed to the User Services Manager, Code 0141 (In-133), ext. 2752 (or ext. 2573 for messages).

The Center provides batch-processing service under OS/VS2 MVS with JES3 and timesharing service under IBM VM/SP CMS. These services are based on an IBM 3033 Attached Processor System with 16 megabytes of storage.

DISTRIBUTION

List 3, plus 1-A5, 300-B2, 3-B3, 1-B13, 3-F3, 1-F4, 1-F5, 1-F6, 1-Code 49, 60-Student Mail Center (Lobby)